

디스플레이 패널의 화질검사를 위한 컨택 지그 위치보정 구현

김성연*, 김영형**, 이종목***, 이용환****

Implementation of Contact-jig Position Correction for Quality Inspection of Display Panels

Seong-Yeon Kim*, Young-Hyung Kim**, Jong-Mok Lee***, and Yong-Hwan Lee****

이 연구는 2020년 국립대학 육성사업비로 지원되었음

요 약

디스플레이 화질검사를 위한 비전 검사의 속도와 정확성 증가를 위해 디스플레이 생산 공정에서 컨택 지그 틀어짐에 대한 고속 검출과 위치 보정이 필요하다. 컨택 지그란 OLED 패널의 성능과 구동의 검사를 위한 물리적 컨택 유닛이다. 패널 지그에 컨택 시, 검사할 디스플레이의 위치가 맞지 않으면 정밀 검사가 어려워 디스플레이의 수평을 맞추는 과정이 필수적이다. 본 연구에서는 이를 위해 패널 틀어짐에 대한 보정 알고리즘을 고안하고 FPGA 영상처리를 통해 현재 검사하는 액정의 위치와 이동해야 할 값을 계산한다. 이미지 전처리 과정은 이미지 잡음 제거와 에지 검출 마스크 필터를 이용한 코너 좌표 검출을 포함한다. 각도 검출 알고리즘 과정은 병렬처리를 함으로써 고속 보정이 가능하도록 하며, 이는 연산 속도를 높이고 패널 생산 증대에 기여할 수 있다.

Abstract

In order to increase the speed and accuracy of vision inspection for display quality inspection, high-speed detection of contact jig distortion and position correction in the display production process are required. The contact jig is a physical contact unit for inspection of the performance and operation of OLED panels. When contacting the panel jig, if the position of the display to be inspected is not correct, it is difficult to perform detailed inspection, and thus the process of leveling the display is essential. In this study, a correction algorithm for panel distortion is devised for this purpose, and the position of the currently inspected liquid crystal and the value to be moved are calculated through FPGA image processing. Image pre-processing in this process includes image noise removal and corner coordinate detection using an edge detection mask filter. The angle detection algorithm enables high-speed correction by performing parallel processing, which can increase the computational speed and increase the panel production.

Keywords

image processing, median filter, sobel mask filter, corner detection, euclidean distance, angle measurement, FPGA

* 금오공과대학교 전자공학과 석사과정
- ORCID: <http://orcid.org/0000-0002-1861-9288>
** 금오공과대학교 IT융합학과 부교수
- ORCID: <https://orcid.org/0000-0001-5708-854X>
*** 위드시스템(주) 경영지원본부 이사
- ORCID: <https://orcid.org/0000-0001-5252-0129>
**** 금오공과대학교 전자공학과 교수
- ORCID: <https://orcid.org/0000-0003-1222-8283>

• Received: Jan. 29, 2021, Revised: Apr. 26, 2021, Accepted: Apr. 29, 2021
• Corresponding Author: Yong-Hwan Lee
Dept. of Electronic Engineering of Kumoh National Institute of Technology, 61 Daehak-ro, Gumi-si, Gyeongbuk, Korea,
Tel.: +82-54-478-7432, Email: yhlee@kumoh.ac.kr

I. 서론

디스플레이 산업이 발전함에 따라 생산량이 증가하면서, 연관 생산 및 검사 기술의 개발이 중요시되고 있다. 디스플레이 화질검사 시 패널이 수평을 이루지 않으면 명확한 영상을 제공할 수 없고 이는 제품 생산의 정확도에 치명적인 영향을 줄 수 있다. 이런 문제를 해결하기 위해 패널의 기울기를 검출하여 생산 공정이 원활하게 진행될 수 있도록 하는 기술이 요구된다. 본 논문에서는 디스플레이 화질검사의 정확도와 속도를 증가시키기 위해 디스플레이 비틀림에 대한 고속 보정을 구현하도록 한다.

그림 1은 패널이 고정된 이미지로, 패널을 제외한 배경은 검은색이며 디스플레이에 전원을 인가하여 색을 비춘다. 패널 검사 시 디스플레이를 고정하는 과정에서 일어날 수 있는 비틀어짐 과정을 해결하기 위해 패널 비틀림 검사 알고리즘을 개발한다.

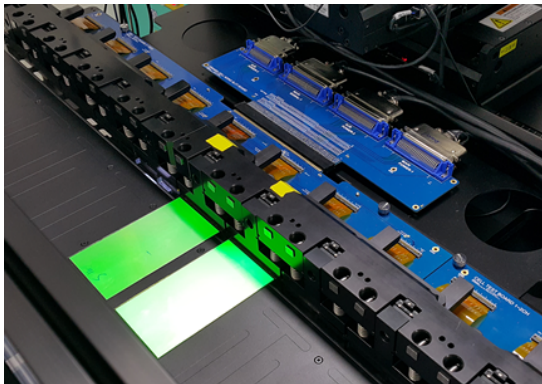


그림 1. 디스플레이 패널
Fig. 1. Display panel

전체 시스템 순서도는 그림 2와 같다. 패널의 틀어진 각도를 검출하기 위해서는 입력된 이미지에 전처리 과정을 거쳐 코너 좌표를 검출하는 과정이 필요하다. 전처리 과정은 입력받은 컬러 이미지를 그레이스케일을 통해 흑백 이미지로 변환하고 미디언 필터를 이용하여 이미지 잡음을 제거, 이후 소벨 필터를 사용하여 디스플레이 에지를 검출하는 과정에서 얻은 결과를 이용해 코너 좌표를 찾는 과정으로 이루어져 있다. 전처리 과정을 통해 검출한 코너 좌표를 디스플레이 비틀어짐 검출 알고리즘에 대입하여 OLED 패널이 기울어진 각도를 찾는다.

II. 전처리 과정

전처리 과정에서 사용된 이미지 크기는 시뮬레이션 과정을 고려하여 640×360 이미지를 사용해 연산한다. 일반적으로 사용되는 컬러 이미지의 경우 한 픽셀당 R, G, B 각각 8-bit 데이터로 구성된 24-bit true 컬러를 사용한다. 그러나 이는 픽셀 단위 처리 시 연산 시간을 많이 소모하고, 이를 저장하기 위한 메모리 크기 또한 커지는 단점이 있어 그레이 스케일(식 (1))[1]로의 변환 과정을 거친다.

$$gray = red * 0.299 + green * 0.587 + blue * 0.114 \quad (1)$$

그레이 스케일 변환 과정을 거친 이미지에는 잡음 제거를 위해 미디언(Median) 필터를 사용하였다. 미디언[2]-[4] 필터는 입력된 이미지의 마스크 값 안에 포함된 픽셀값을 오름 또는 내림차순으로 정렬한 후 중간값을 사용하는 방법이다.

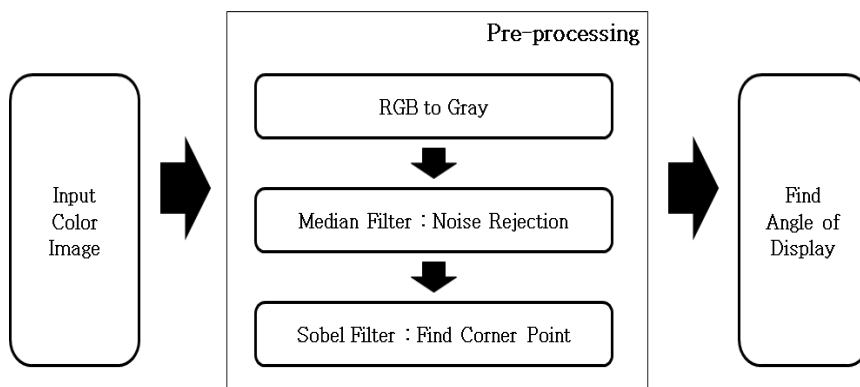
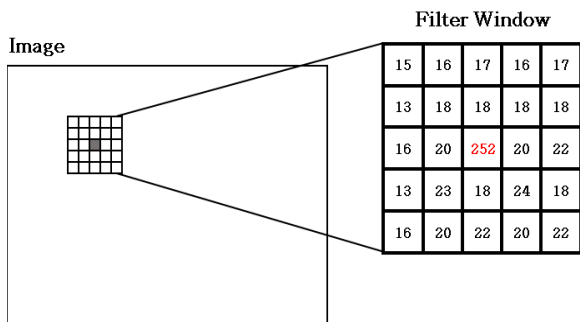


그림 2. 시스템 순서도
Fig. 2. System flowchart

사용한 5x5 마스크 크기의 미디언 필터 예시는 그림 3에 나타난다. 미디언 필터는 주변과 비교하여 극단적으로 농도 차이를 보이는 값은 정렬 시 왼쪽 혹은 오른쪽으로 치우쳐 중앙값이 될 수 없다. 이런 이유로 미디언 필터는 잡음 제거에 뛰어나다. 중간 값을 검출하였다면, 블러링에 따른 이미지의 번짐 제거를 위해 필터 결과가 특정 범위에 따라 0과 255, 흑백으로 출력한다. 이후 결과는 소벨[5][6] 필터를 이용하여 에지를 검출하고, 그 결과를 이용해 코너 좌표를 구한다.

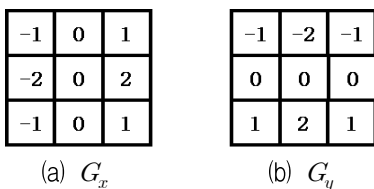
가로 방향 미분 계산 마스크와 세로 방향 미분 계산 마스크 두 개로 이루어지며, 3x3 크기의 소벨 마스크는 그림 4에 나타내었다. 소벨 마스크는 미분 값을 이용해 영상의 경계선을 추출하며, 이는 대부분 색 또는 밝기가 급격하게 변하는 구조를 가진다. 코너 좌표 검출은 에지 검출과정과 동시에 이루어진다. 3x3 소벨 마스크의 가중치를 이용하여 구한 가로와 세로의 필터 결과의 합 $G_s(2)$ 를 이용한다. 미디언 필터 결과값을 0과 255 두 값 중 하나로 출력하였기에 코너 좌표[7][8]의 G_s 값을 쉽게 계산할 수 있다.



Pixel Alignment

13 13 15 16 16 ... 18 18 ... 22 23 24 252
Median Value Noise

그림 3. 5x5 미디언 필터 마스크
 Fig. 3. 5x5 median filter mask



(a) G_x (b) G_y

그림 4. 소벨 마스크 필터
 Fig. 4. Sobel mask filter

이를 이용하여 코너 점이 될 수 있는 픽셀값의 범위를 특정하도록 한다.

$$G_s = \sqrt{G_x^2 + G_y^2} = |G_x + G_y| \quad (2)$$

소벨 마스크 필터의 결과가 코너 좌표로 인식되는 픽셀값의 범위에 들어올 때 그 좌표를 코너 점으로 인식하고, 인식된 코너 점들을 비교하여 각도 검출에 사용할 두 개의 코너 좌표만을 검출한다. 픽셀이 코너 좌표, 에지, 혹은 두 경우에 포함되지 않는 경우마다 다른 G_s 값을 가진다.

그림 5의 경우 코너 좌표에서는 6W, 코너를 제외한 에지에서는 4W의 결과를 가진다. 이 경우 코너 좌표를 검출하기 위해서는 6W를 기준으로 오차를 둔 범위에 포함되는 좌표를 코너로 인식하도록 한다. 4개의 코너 좌표 중 상단 2개의 코너 값만 이용하여 패널이 틀어진 각도를 측정할 것이기에, 조건문을 주어 상단의 두 좌표만 검출한다. 소벨 마스크 필터를 이용한 디스플레이의 코너 검출 결과는 그림 6에 나타내었다.

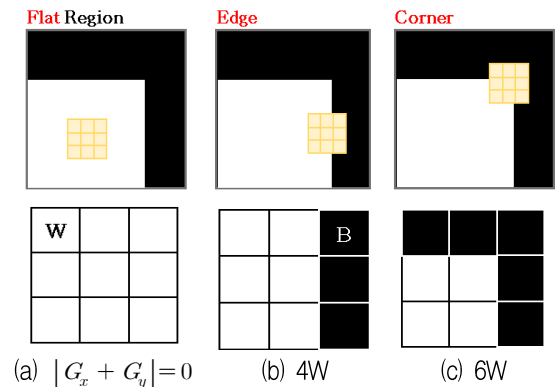


그림 5. 소벨 필터를 이용한 코너 검출
 Fig. 5. Corner detection using Sobel filter

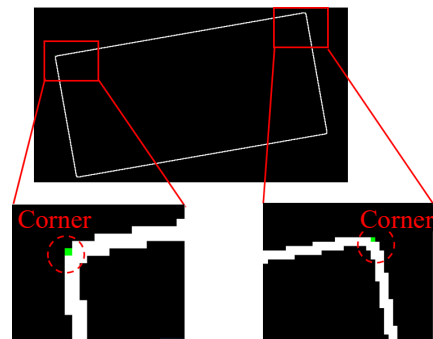


그림 6. 코너 검출
 Fig. 6. Corner detection

III. 각도 검출

전처리 과정을 통해 4개의 코너 좌표를 검출하였다면 유클리디안 거리 식 (2)을 이용하여 영상의 외곽선 길이를 측정한다.

$$\sqrt{(p_1 - q_1)^2 + (p_2 - q_2)^2 + \dots + (p_n - q_n)^2} \quad (2)$$

$$= \sqrt{\sum_{i=1}^n (p_i - q_i)^2}$$

4개의 코너 좌표 중 상단 2개의 코너 값만 이용하여 패널이 틀어진 각도를 측정한다. 상단 두 개의 코너 중 좌측 코너값을 원점으로 지정 후, 우측 코너값에서 원점의 좌표값을 빼주면 새로운 좌표를 얻을 수 있다.

그림 7과 같이 얻어낸 새로운 두 좌표의 x값, y값 차이를 역삼각함수 식 (3)에 대입하여 디스플레이의 틀어진 각도를 구할 수 있다.

$$\theta = \tan^{-1}\left(\frac{y}{x}\right) \quad (3)$$

FPGA 구현 시, 좌표 범위가 넓기 때문에 IP 코어를 사용하여 구현하더라도 각도의 계산이 오래 걸린다. 계산 속도를 줄이기 위해 Look-Up-Table (LUT)[9]을 사용하여 구현한다. 이와 같은 방법으로 패널의 틀어진 각도를 구하는 테스트를 반복하여 결과의 정확도를 확인한다.

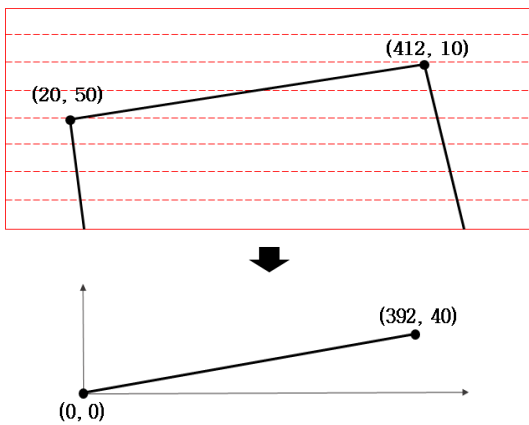


그림 7. 새로운 좌표 계산
Fig. 7. Calculation of new coordinates

디스플레이 패널의 틀어진 오차 범위는 2.0° 이 내로, 이를 초과할 시 검사장비에 삽입되지 않는 구조를 가진다. 만일 각도 검출을 위해 계산한 삼각함수 공식 대입 값이나 각도 검출 결과가 적정 범위를 벗어나게 되면 잘못된 접근이라는 경고를 띄운다.

표 1은 실제 틀어진 값과 계산된 값의 오차를 표시한 것이다. 평균 오차는 0.076°로 나타나며, 계산 오차는 0.2°를 넘지 않는 것을 확인할 수 있다.

표 1. 테스트 결과
Table 1. Test result

Actual angle (degree, °)	Calculated angle (degree, °)	Error (degree, °)
0.00	0.00	0.00
0.10	0.11	-0.01
0.20	0.22	-0.02
0.30	0.33	-0.03
0.40	0.43	-0.03
0.50	0.54	-0.04
0.60	0.65	-0.05
0.70	0.76	-0.06
0.80	0.87	-0.07
0.90	0.98	-0.09
1.00	1.09	-0.09
1.10	1.20	-0.10
1.20	1.31	-0.11
1.30	1.42	-0.12
1.40	1.53	-0.13
1.50	1.64	-0.14
1.60	1.75	-0.15
1.70	1.86	-0.16
1.80	1.86	-0.06
1.90	1.97	-0.07
2.00	2.08	-0.08

IV. FPGA 구현

FPGA 설계의 경우 XILINX의 7-series 보드를 사용하여 구현하였으며, 전 과정의 데이터를 저장하지 않고 가로 열 라인 버퍼를 사용하여 1줄에 대한 데이터만 저장하여 메모리 사용을 최소화 하였다.

라인 버퍼[10][11]는 첫 번째 버퍼에 데이터가 입력되고, 입력이 완료되면 순차적으로 두 번째, 세 번째 버퍼에 데이터가 입력된다. 마지막 버퍼에 데이터가 입력됨과 동시에 이전 버퍼에 저장된 데이터들은 마스크 연산을 실행한다. 마지막 버퍼의 데

이터 입력이 끝나면 첫 번째 버퍼에 새로운 데이터가 쓰이며, 이와 동시에 두 번째, 세 번째, 네 번째 버퍼에 데이터들은 마스크 연산을 실행한다. 위 과정은 버퍼가 돌아가며 진행, 세로 열의 마지막 데이터가 입력될 때까지 반복된다. 가로 열 라인 버퍼와 파이프라인 적용, 병렬처리[12] 설계를 적용하여 데이터 처리 속도를 증가시켰다. 전반적인 FPGA 설계 순서는 그림 8에서 확인할 수 있다.

그림 9는 소벨 마스크 필터의 과정에서 사용한 가로열 라인 버퍼이다. 가운데 숫자는 버퍼 넘버를 나타내며, 첫 번째 버퍼에 데이터 입력이 완료되면 순차적으로 다음 버퍼에 데이터가 입력된다. 전처리를 거친 데이터는 순차적으로 소벨 필터의 라인 버퍼에 입력된다.

그림 10과 같이 하나의 라인 버퍼에 입력 완료 시 다음 라인 버퍼에 입력되며, 3개의 라인 버퍼 입력이 완료되면 4번째 라인 버퍼 입력이 시작될 때 기존 3개의 라인 버퍼에 대한 3x3 마스크 소벨 필

터의 결과가 나온다.

미디언 필터의 경우 5x5 마스크를 사용하였기에 6줄의 라인 버퍼가 필요하며, 사용 방법은 소벨 필터와 동일하다. 미디언 필터의 결과는 6번째의 라인 버퍼 입력이 시작될 때 순차적으로 출력되어 소벨 마스크 필터 검출기의 첫 번째 라인 버퍼의 입력으로 사용된다.

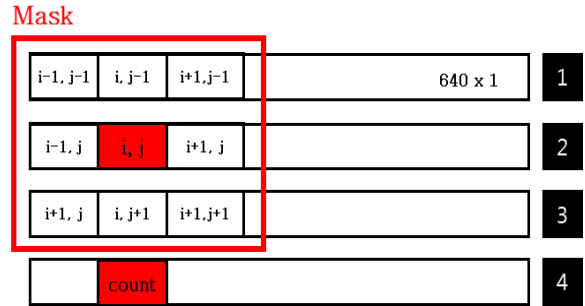


그림 10. 라인 버퍼와 소벨 마스크 필터
Fig. 10. Line buffer and sobel mask filter

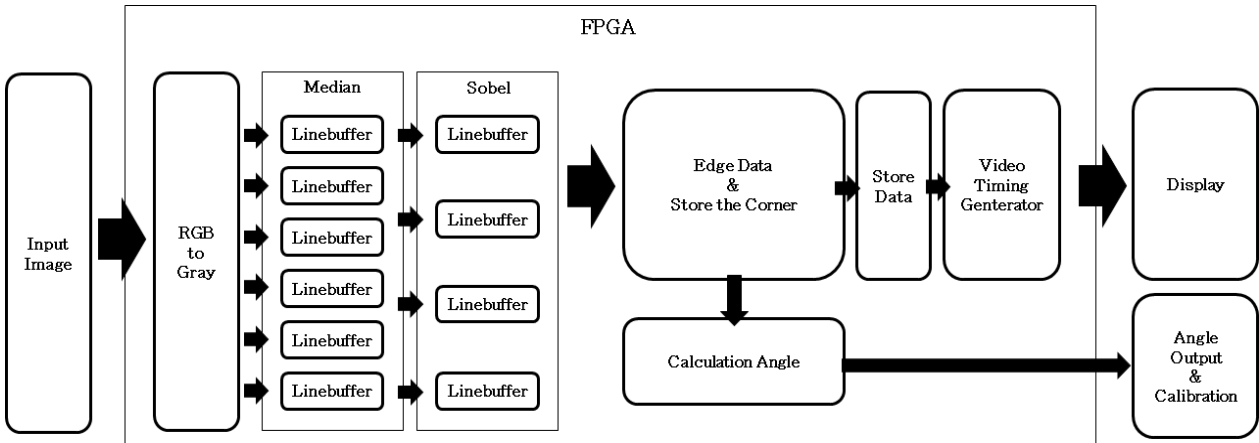


그림 8. FPGA 설계 구조도
Fig. 8. Structure of FPGA design

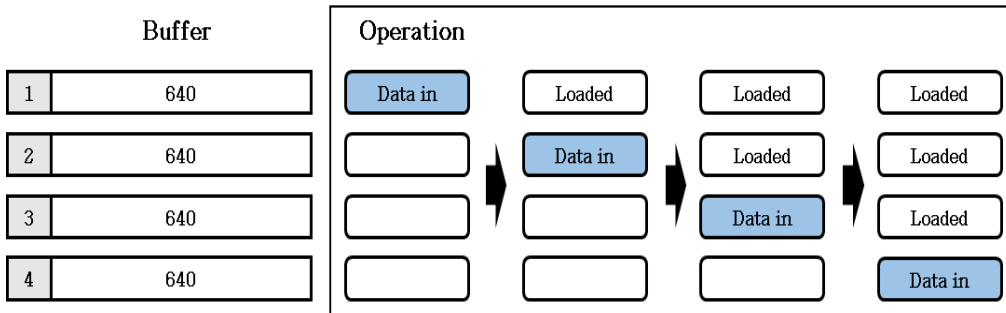


그림 9. 가로 열 라인 버퍼
Fig. 9. Horizontal column line buffer

V. 실험 결과

소프트웨어의 경우 단계별로 데이터를 호출하고 연산하는 구조로, 모델링 시 각각의 단계별로 메모리를 저장하기에 그레이스케일, 미디언 필터, 소벨 마스크 필터 사용을 위해 동일 크기의 메모리 3개가 소모되는 구조를 가진다. 이는 FPGA 설계 시 병렬처리를 통해 기본 소모 clock을 감소시키고, 가로 열 라인 버퍼를 사용함으로써 데이터 소모를 줄일 수 있다. 본 논문의 FPGA 설계에서는 가로 열 라인 버퍼를 사용하여 흑백 데이터는 저장하지 않는다. 또한, 미디언 필터와 소벨 필터의 입력 값을 저장하는데 미디언 필터의 경우 640×1 크기의 라인 버퍼 6개, 소벨 필터의 경우 4개씩만 사용한다. 그림 11은 소프트웨어와 FPGA 설계 구현의 차이를 보여준다.

표 2는 소프트웨어와 FPGA의 clock과 메모리 차이를 나타낸다. Software 구현의 경우 전처리 과정에서 많은 클럭을 소모하고, 메모리 또한 그레이스케일, 미디언 필터, 소벨 필터 과정에서 각각 57,600의 메모리 크기를 소모한다. 이는 필터 과정이 동시에 진행되는 FPGA와 비교할 때 매우 큰 차이를 보인다. FPGA 구현 결과 전체 전처리 과정에서 231,680의 clock을 소모하며, 메모리 또한 그레이스케일과 미디언 필터의 잡음 제거 과정에서 크게 줄일 수 있다. FPGA 설계의 합성 결과는 그림 12와 표 3에 나타내었다.

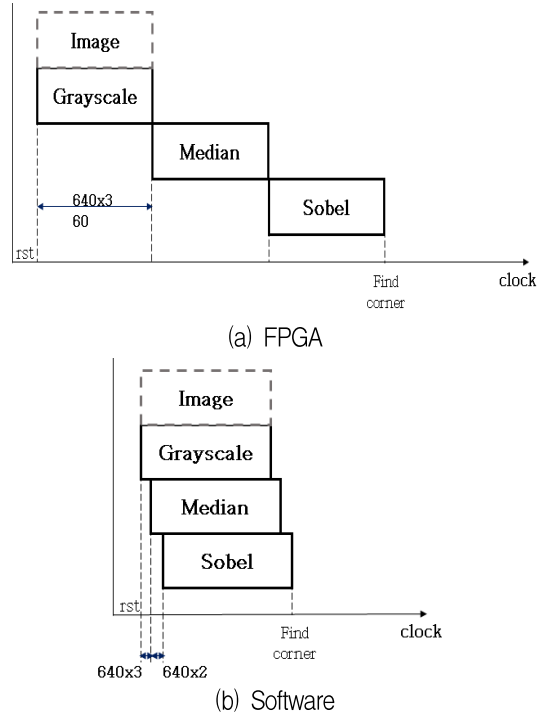


그림 11. 소프트웨어와 FPGA 설계의 비교

Fig. 11. Comparison of software and FPGA design

표 2. 알고리즘 비교

Table 2. Algorithm comparison

		Software	FPGA
clock	RGB to gray	230,400	-
	Median	2,073,600	231,680
	Sobel	2,073,600	
Memory	RGB to gray	57,600	-
	Median	57,600	3840
	Sobel	57,600	2560 57,600

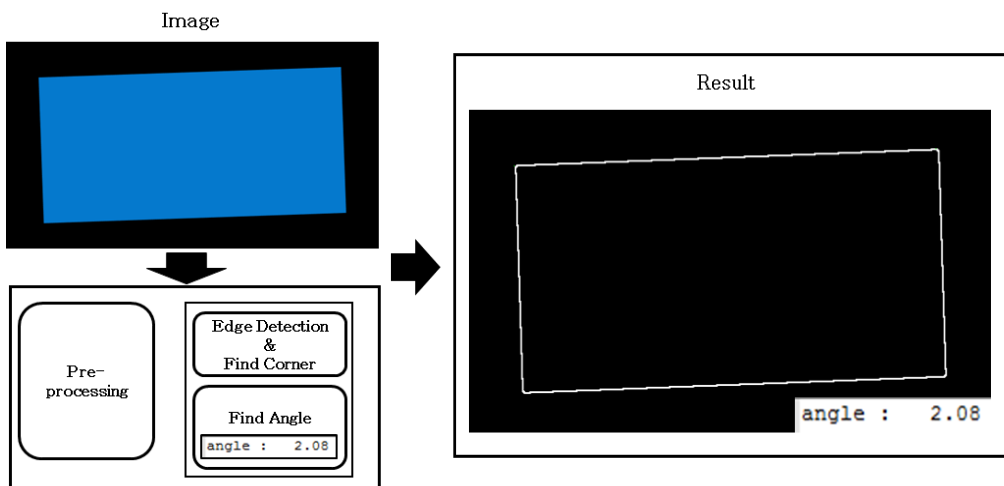


그림 12. FPGA 시뮬레이션

Fig. 12. FPAG simulation

표 3. FPGA 합성 결과

Table 3. FPGA synthesis result

	Used cells	Allowed cells	Utilization(%)
Device	xc7v330tffg1157-3		
LUT	107,771	204,000	52.83
FF	33,795	408,000	8.28
Block ram(36Kb)	155.50	750	20.73

VI. 결론

본 논문에서는 디스플레이 틀어짐을 자동으로 고속 보정이 가능하도록 하는 하드웨어를 개발하여 고속 검사를 할 수 있도록 하였다. 비틀린 각도 검출 알고리즘을 이용해 디스플레이 틀어짐으로 인한 비전 검사 시 에러 발생률을 감소시키고, 라인 버퍼와 파이프라인을 이용해 구현하여 데이터 메모리 소모를 줄인다. FPGA 구현 시, 소프트웨어와는 달리 전처리 과정에서 231,680 clock을 소모하고 메모리 크기 또한 감소시킬 수 있으며, 각도 보정에 따른 오차 또한 평균오차 0.076°로 정확하게 나타난다.

본 논문을 적용하여 디스플레이 생산 라인 검사 공정의 시간 단축과 불량률 축소를 가능하게 하고, 이는 생산성 향상 및 품질 향상으로 이어질 것이다.

References

- [1] Ahmad Ijaz, "Color-to-Grayscale Algorithms Effect on Edge Detection - a Comparative Study", Electronics, Information, and Communication (ICEIC), International Conference, Honolulu, HI, USA, pp. 1-4, Jan. 2018. <https://doi.org/10.23919/ELINFOCOM.2018.8330719>.
- [2] G. George, R. M. Oommen, S. Shelly, S. S. Philipose, and A. M. Varghese, "A Survey on Various Median Filtering Techniques For Removal of Impulse Noise From Digital Image", Conference on Emerging Devices and Smart Systems (ICEDSS), Tiruchengode, India, pp. 235-238, Mar. 2018. <https://doi.org/10.1109/ICEDSS.2018.8544273>.
- [3] Zhang Xin Ming, Kang Qiang, Cheng Jin Feng, and Wang Xia, "Adaptive Four-dot Median Filter for Removing 1-99% Densities of Salt-and-Pepper Noise in Image", Journal of Information Technology Research, Vol. 11, No. 3, pp. 47-61, Sep. 2018. <http://dx.doi.org/10.4018/JITR.2018070104>.
- [4] A. Furnari, G. M. Farinella, A. R. Bruna, and S. Battiato, "Generalized Sobel Filters for gradient estimation of distorted images", 2015 IEEE International Conference on Image Processing (ICIP), Quebec City, Canada, pp. 3250-3254, 2015. <https://doi.org/10.1109/ICIP.2015.7351404>.
- [5] Jin-Su Kang, Sung-Soo Kim, Yong-Hwan Lee, and Young-Hyung Kim, "Vision Inspection and Correction for DDI Protective Film Attachment", Journal of Advanced Information Technology and Convergence, Vol. 10, No. 2, pp. 153-166, Dec. 2020. <https://doi.org/10.14801/JAITC.2020.10.2.153>.
- [6] Z. Li and J. Wang, "An Adaptive Corner Detection Algorithm Based on Edge Features", 2018 10th International Conference on Intelligent Human-Machine Systems and Cybernetics (IHMSC), Hangzhou, China, pp. 191-194, 2018. <https://doi.org/10.1109/IHMSC.2018.10150>.
- [7] Seong-Yeon Kim, Yong-Hwan Lee, and Hee-Jin Lee, "Slope Calibration Algorithm for OLED Display Gamma-Tuning Instruments", JKIIIT, Vol. 18, No. 9, pp. 43-50, 2020. <https://doi.org/10.14801/jkiit.2020.18.9.43>.
- [8] Seong-Yeon Kim, "Development of Algorithm for Measuring Slope of Objects", M. S. dissertation, Department of Electronic Engineering, Kumoh National Institute of Technology, Gumi, Korea, 2020.
- [9] In-Gook Chun, "Printable Image Watermarking Based on Look-Up Table", Journal of the Korea Academia-Industrial cooperation Society, Vol. 7, No. 4, pp. 656-664, Aug. 2006.
- [10] H. Wang, T. Wang, L. Liu, H. Sun, and N. Zheng, "Efficient Compression-Based Line Buffer Design for Image/Video Processing Circuits," in IEEE Transactions on Very Large Scale Integration (VLSI) Systems, Vol. 27, No. 10, pp.

2423-2433, Jun. 2019. <https://doi.org/10.1109/TVLSI.2019.2921249>.

[11] Chan-Su Park and Hi-Seok Kim, "FPGA Implementation for Real Time Sobel Edge Detector Block Using 3-Line Buffers", Journal of IKEEE, Vol. 19, No. 1, pp. 10-17, Mar. 2015. <https://doi.org/10.7471/ikeee.2015.19.1.010>.

[12] S. H. Jin, J. U. Cho, K. H. Kwon, and J. W. Jeon, "An FPGA Implementation of Parallel Hardware Architecture for the Real-time Window-based Image Processing", Journal of KIPS Transactions, Vol. 13B, No. 106, pp. 233-230, Jun. 2006. <https://doi.org/10.3745/KIPSTB.2006.13B.3.223>.

저자소개

김 성 연 (Seong-Yeon Kim)



2019년 2월 : 금오공과대학교
전자공학부(공학사)
2019년 2월 ~ 현재 : 금오공과
대학교 전자공학과(공학석사)
관심분야 : Soc, 인터페이스,
이미지처리

김 영 형 (Young-Hyung Kim)



1992년 : 금오공과대학교
전자공학부(공학사)
2010년 : 금오공과대학교
산업경영학과(박사)
2017년 ~ 현재 : 금오공과대학교
IT융합학과 교수
관심분야 : 전기 및 전자 산업 시스템

이 종 목 (Jong-Mok Lee)



2015년 : 구미대학교
산업경영학과(학사)
2020년 : 금오공과대학교
테크노경영학과(석사)
2008년 ~ 현재 : 위드시스템(주)
경영지원팀 이사
관심분야 : 자동차 설비 장치 및
OLED 디스플레이 검사

이 용 환 (Yong-Hwan Lee)



1993년 : 연세대학교
전자공학과(공학사)
1999년 : 연세대학교
전자공학과(공학박사)
1999년 ~ 2004년 : 삼성전자,
하이닉스반도체
2004년 ~ 현재 : 금오공과대학교

전자공학부 교수

관심분야 : SoC, 컴퓨터 비전, 고속인터페이스,
임베디드시스템, 마이크로프로세서 아키텍처